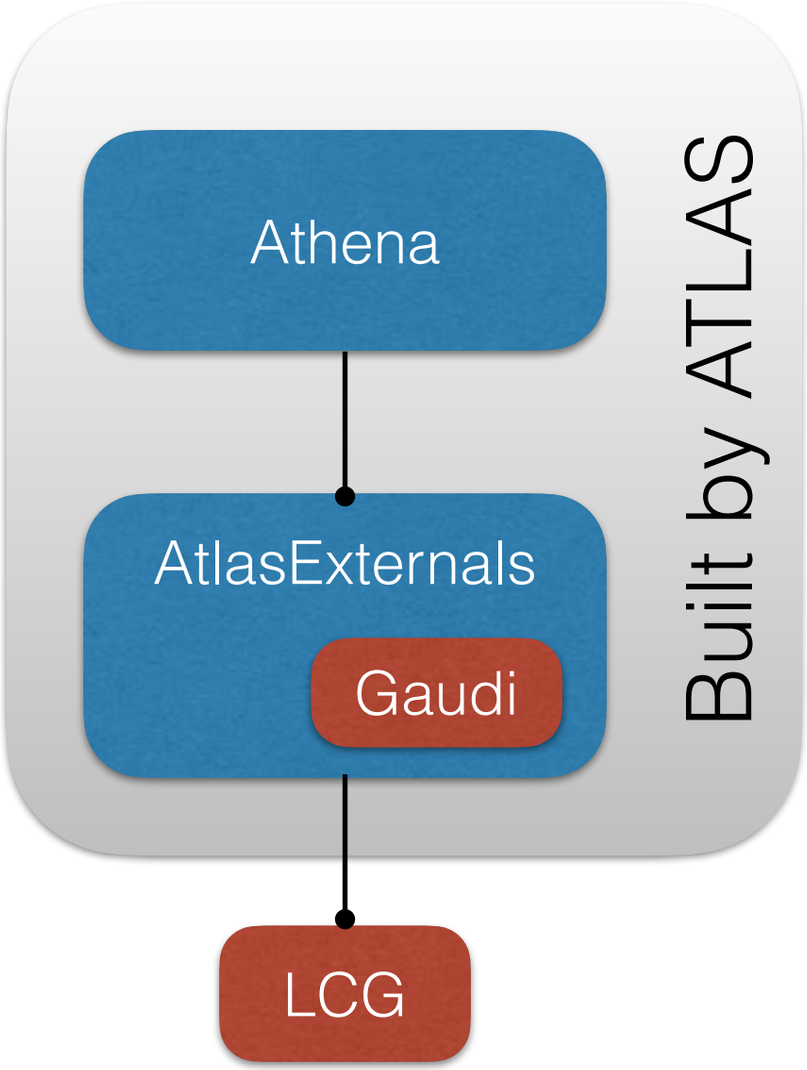
HEPScore & ATLAS

Walter Lampl
University of Arizona

The ATLAS software stack

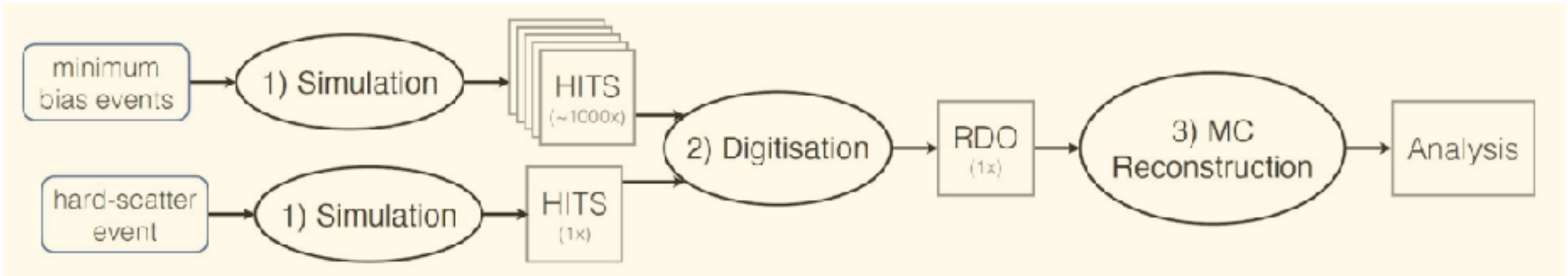
- Based on LCG: ROOT, Geant4, Generators, ….
- Some more externals that we build ourselves: Ex COOL, also few packages that are also in LCG
- The basic framework is Gaudi, shared with LHCb
- atlas/athena, our proper code-base
- Projects:
  - In atlas-parlance, a project is a subset of the code-base built for a specific purpose
  - Example: AthGeneration, AthSimulation
  - Athena is the superset of everything, used for Reconstruction



The processing chains

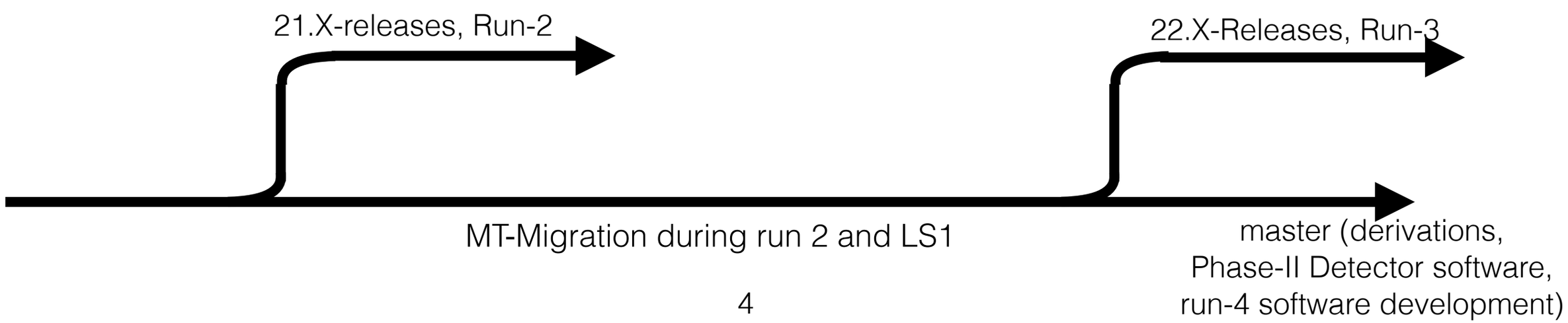
- The framework allows for a lot of flexibility and the workflows got re-optimized few time over the last years
    - Moreover one grid job can entail multiple, chained athena jobs with very different resource usages, the job-transforms can chain athena jobs to produce the desired output out of a given input
    - We expect that the run-3 scenario remains valid until we start run-4 production in 2027/8
- Event Generation is one separate step (Note: Large zoo of generators)
- Detector-simulation is one separate step (either Geant4 or Fast-simulation)
- Digitization and Pile-up-mixing is one step, usually including also Trigger and Reconstruction
- Real-data reconstruction used to be also multi-step because of the memory consumption of DQ-Monitoring. Today we use the “RAWtoALL” workflow, eg one athena job produces all required output is in one go
- Then comes derivation and analysis steps ….

	Value	Percent
MC Simulation Full	31.0 Tri	29%
MC Event Generation	21.4 Tri	20%
MC Reconstruction	16.9 Tri	16%
Group Production	14.9 Tri	14%
Group Analysis	11.6 Tri	11%
User Analysis	9.68 Tri	9%
MC Simulation Fast	1.21 Tri	1%
Data Processing	799 Bil	1%



Releases & Release-stability

- Release numbers have usually three digits like 21.0.83
- For Tier0-processing we typically build new release once per week, for simulation the release usually matures after some infancy-issues and we use the same release for a long time
- 21-series: Used for run 2, typically Multi-Processing mode (fork-after-intialize)
- 22-series: Used for run 3, typically in Multi-Threaded mode
- Releases use for some production-workflows (upgrade-simulation, derivations) are actually built out of the master-branch but labelled 22.X.Y



ATLAS workloads in the repository

digi-reco: Run2-style Digi-Reco based on athena version 21.0.77

gen: Run2-style Event Generation based on athena version 19.2.5.5, using POWHEG+Pythia8

gen-sherpa: Event Generation based on athena version 21.6.84, using Sherpa

kv: aka Kit-Validation, really old

reco-mt: Run-3 style real-data reconstruction

sim: Run 2 style Sim (GEANT4) based on athena version 21.0.15

sim-mt: Run 3 style Simulation (GEANT4) based on athena version 22.0.25

ATLAS workloads in the HEPScore basket

digi-reco: Run2-style Digi-Reco based on athena version 21.0.77

gen: Run2-style Event Generation based on athena version 19.2.5.5, using POWHEG+Pythia8

gen-sherpa: Event Generation based on athena version 21.6.84, using Sherpa

kv: aka Kit-Validation, really old

reco-mt: Run-3 style real-data reconstruction

sim: Run 2 style Sim (GEANT4) based on athena version 21.0.15

sim-mt: Run 3 style Simulation (GEANT4) based on athena version 22.0.25

Put in question because of the long run-time

Usage of ARM and GPU

ARM:

- First port of some of our SW stack succeeded in ~ 2017 but stalled once the person driving it left
- Re-started recently since we have now an LCG-stack built for ARM
    - Regular nightly builds, kind-of-working but not tested on a large scale

GPU:

- A GPU-tracking code was developed for the HLT as early as 2012 but abandoned
- Today, GPU (and heterogenous architectures in general) are part of the run-4 R&D work
    - Not used in production at this point except few dedicated analysis jobs

Management-related questions (1/2):

What is the deployment plan of a new benchmark?

- Development 0.5y? Validation 1-1.5y? Pledging: 2y? Ballpark is 3-4years

How sites should contribute?

- They should be providing all the numbers, running the benchmark in each of their machine types, and confirming the numbers that the experiments measure… ATLAS can confirm with HammerCloud

- We believe that some of the HS06 shortcomings come from not running the HS06 frequently enough -> these operational issues should be solved before/in parallel, cause this will affect whatever benchmark

How the pledges and procurement should then start to include HEPscore (in parallel to HS06 at first, to then gradually replace it)

- As an ATLAS Computing Coordinator: use single currency at a time. We need to agree that the e.g. 2026?27? preliminary pledges will be in HEPSCORE22 units (for example), and in 2025 all accounting is in HEPSCORE22 units.

- As a site administrator:I would want that the year N machines are counted in the year N currency, so that in the transition year I don't suddenly have a lot more to buy than the pledge change suggests.

Management-related questions (2/2):

What are the validation procedures foreseen in these phases?

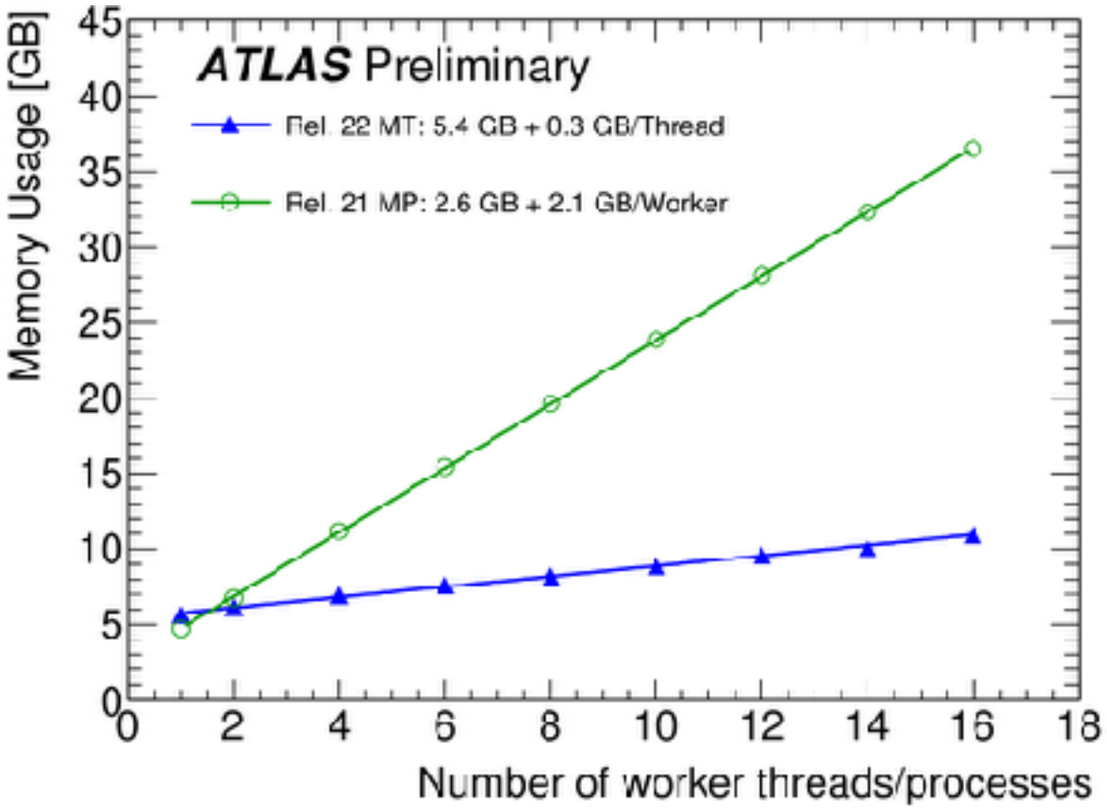
- Site checking their numbers, experiments crosschecking with representative jobs (and/or HC)?

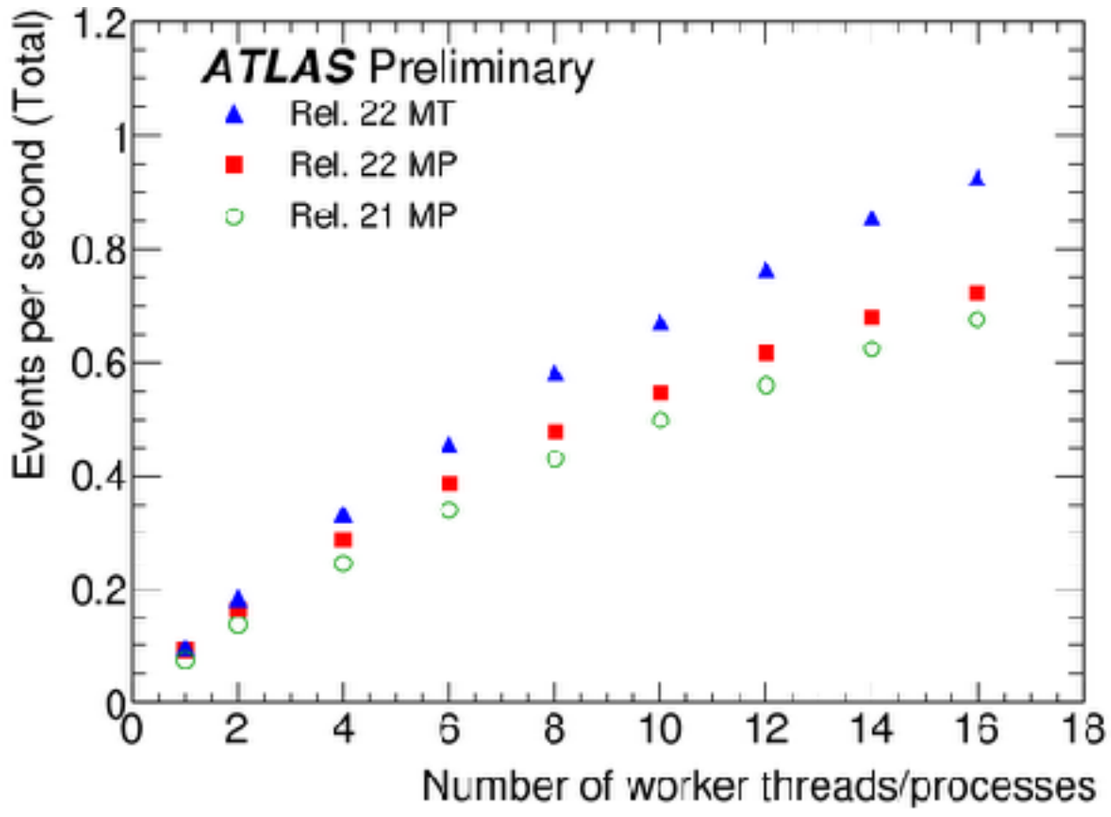
How often a benchmark should/could be changed? every 4-5 years (at each LS?)

- No more often then every 5 years – and probably 10 years is more reasonable. The complexity of changing pledge models has too long a lead time for us.
- Possible target could be to have an update towards the end of LS4.

Backup

MP vs MT Performance





Comparison of Multi-Process vs Multi-Threaded Reconstruction. Memory consumption and throughput